

oops!

00P's not what I thought

**@AnjanaVakil**

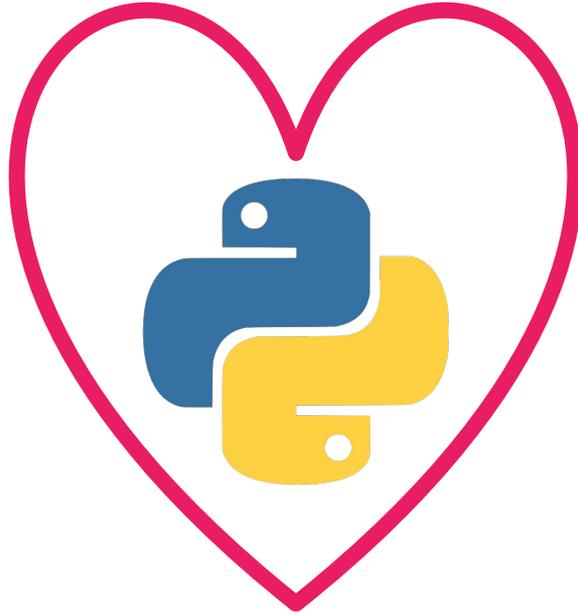
**DomCode 2016**

**hi, I'm Anjana!**

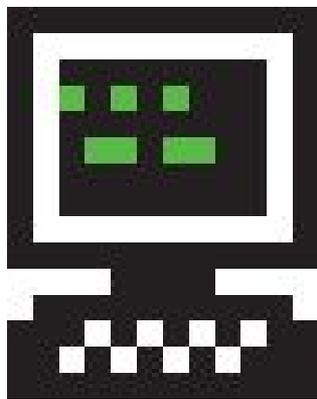
**hi, I'm Anjana!**

**Über**  
**Research**

**hi, I'm Anjana!**



# hi, I'm Anjana!



The Recurse Center

[recurse.com](https://recurse.com)

**hi, I'm Anjana!**

**...and I had OOP all wrong**

# object- oriented programming

objects (!)

classes

inheritance

hierarchy

C++, Java

---

# object- oriented programming

~~objects (!)~~

~~classes~~

~~inheritance~~

~~hierarchy~~

~~C++, Java~~

---

# object- oriented programming



# Alan Kay

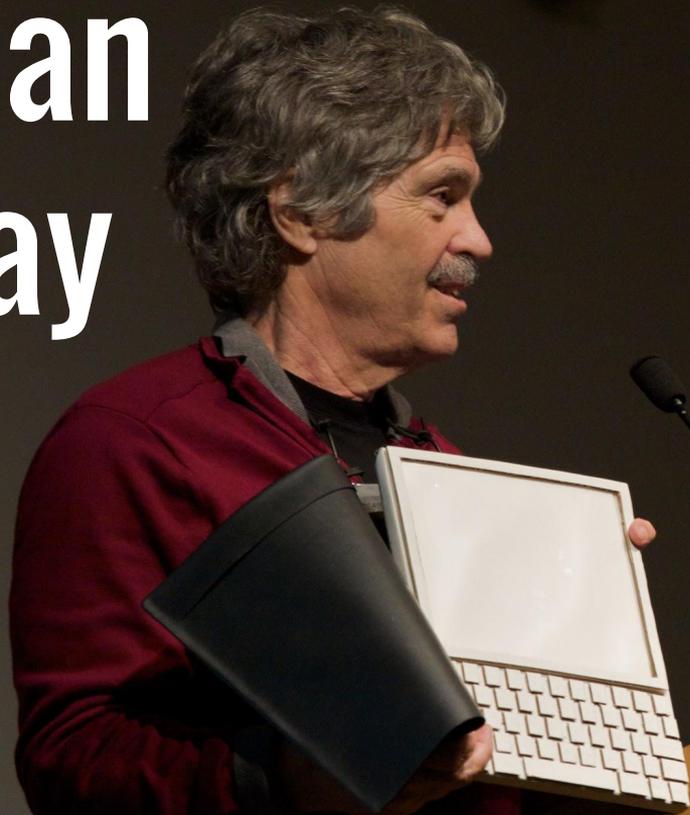
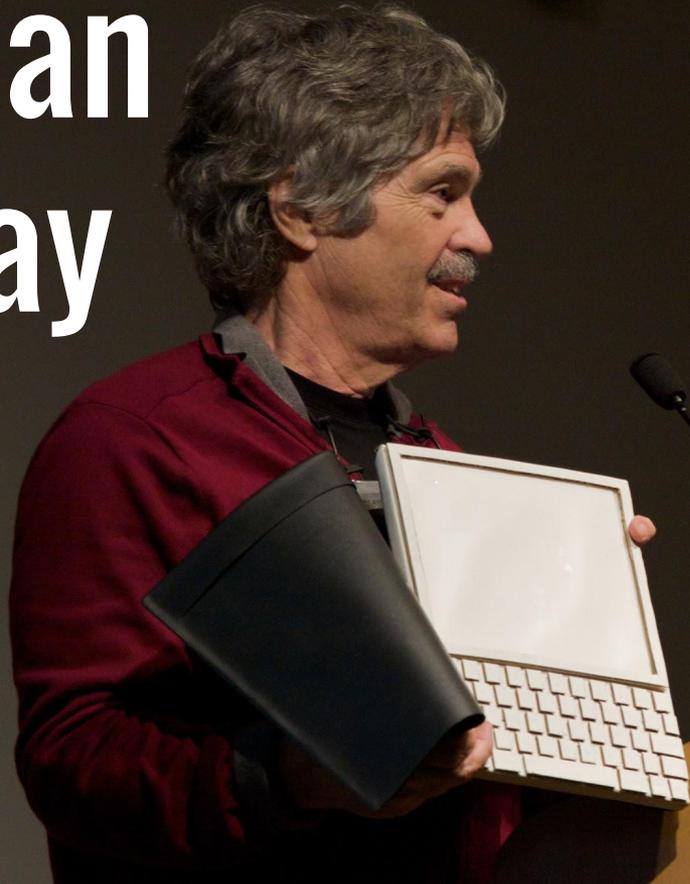


Photo by Marcin Wichary from San Francisco, U.S.A. (Alan Kay and the prototype of Dynabook, pt. 5)  
[CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0>)], via Wikimedia Commons

# Alan Kay

# personal computing pioneer



← **Dynabook**

# Alan Kay

# coiner of “object- oriented”

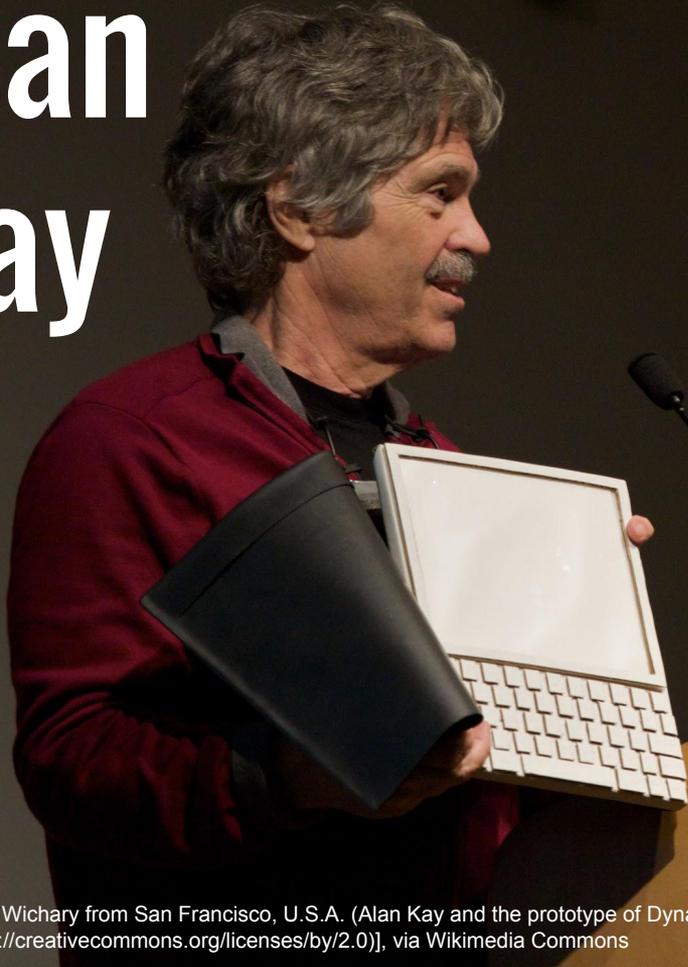


Photo by Marcin Wichary from San Francisco, U.S.A. (Alan Kay and the prototype of Dynabook, pt. 5)  
[CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0>)], via Wikimedia Commons

# Alan Kay

# (co-)creator of Smalltalk

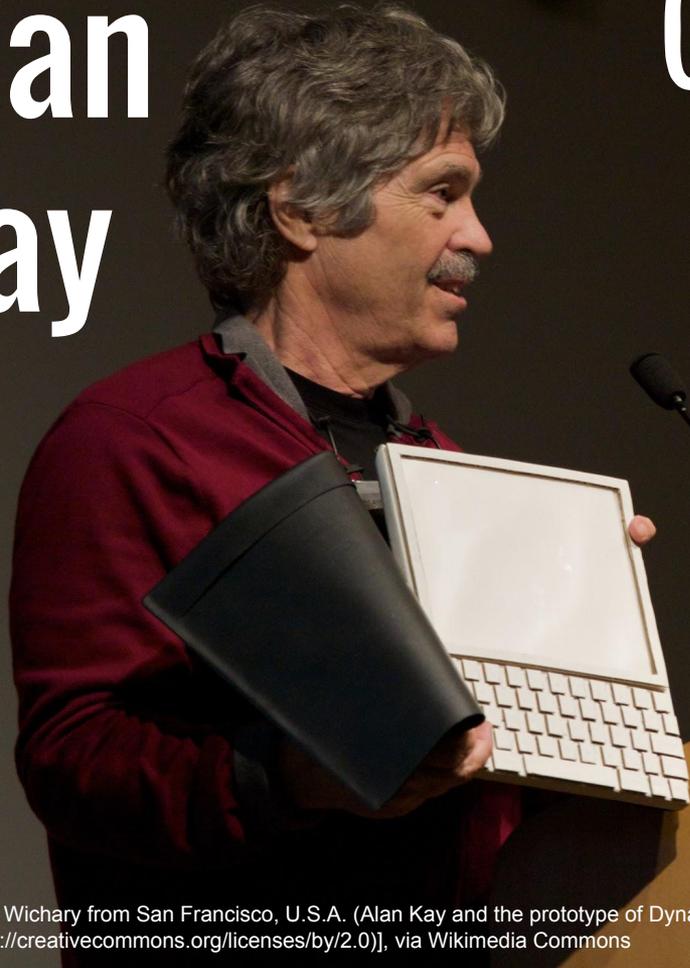


Photo by Marcin Wichary from San Francisco, U.S.A. (Alan Kay and the prototype of Dynabook, pt. 5)  
[CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0>)], via Wikimedia Commons

# Alan Kay

# undergrad biology major

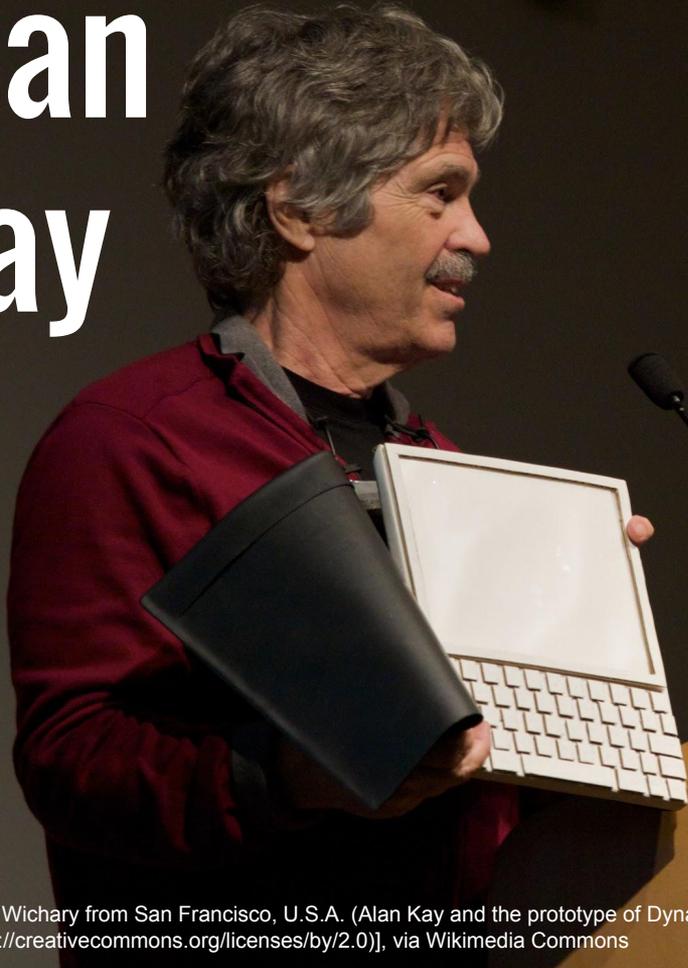


Photo by Marcin Wichary from San Francisco, U.S.A. (Alan Kay and the prototype of Dynabook, pt. 5)  
[CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0>)], via Wikimedia Commons

oops #1:

biological metaphors

**what I  
thought**

**inheritance**

**taxonomy**

---

**what it is**

**cells**

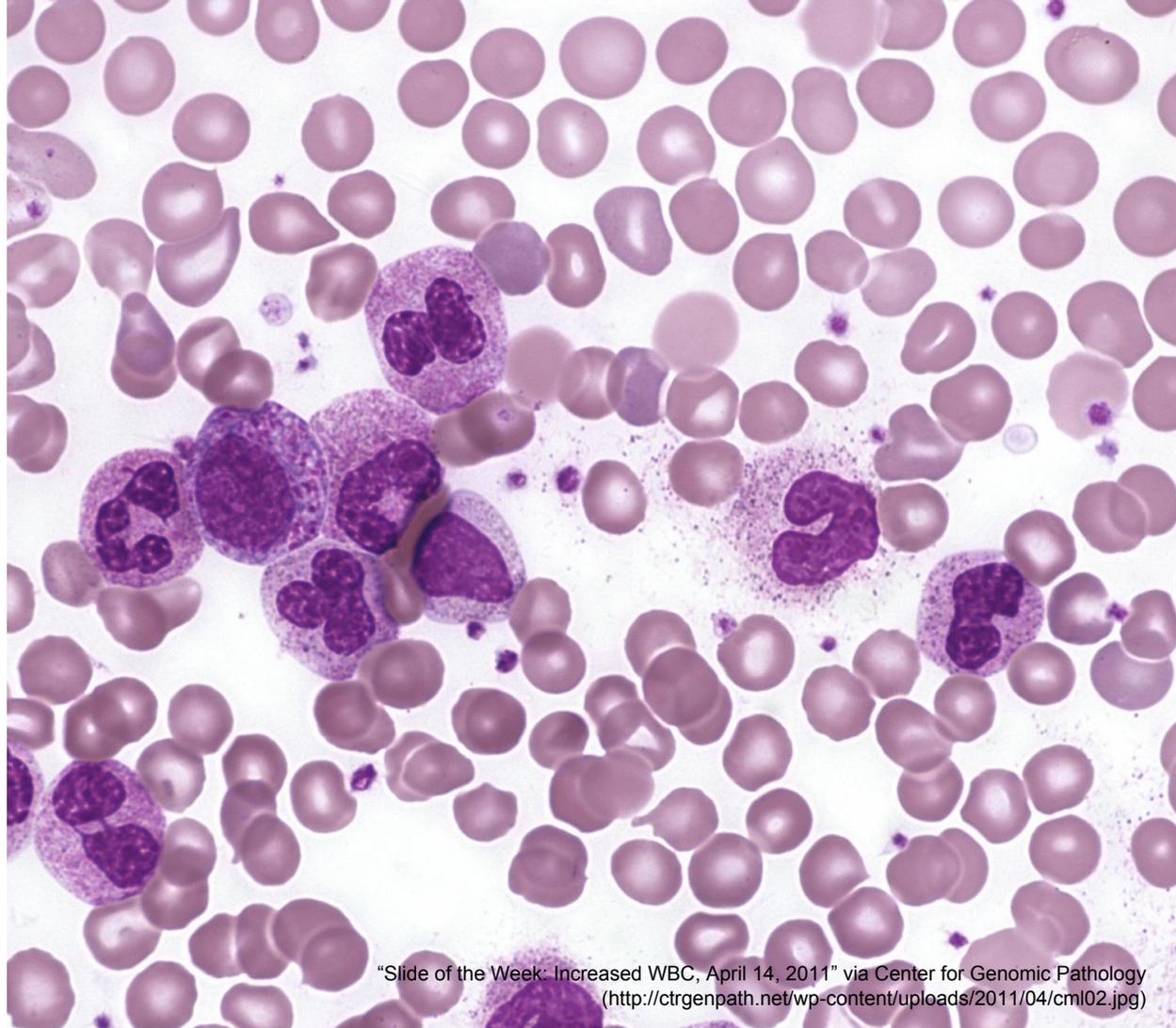
**organs**

---

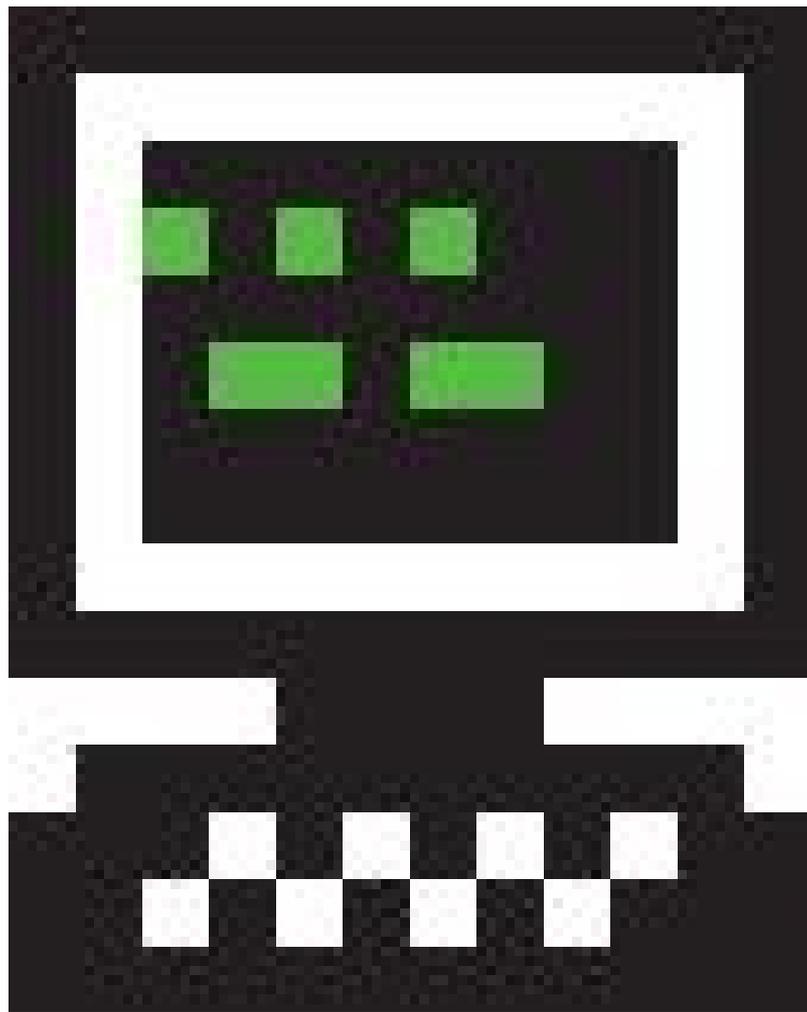
**procedural  
imperative**

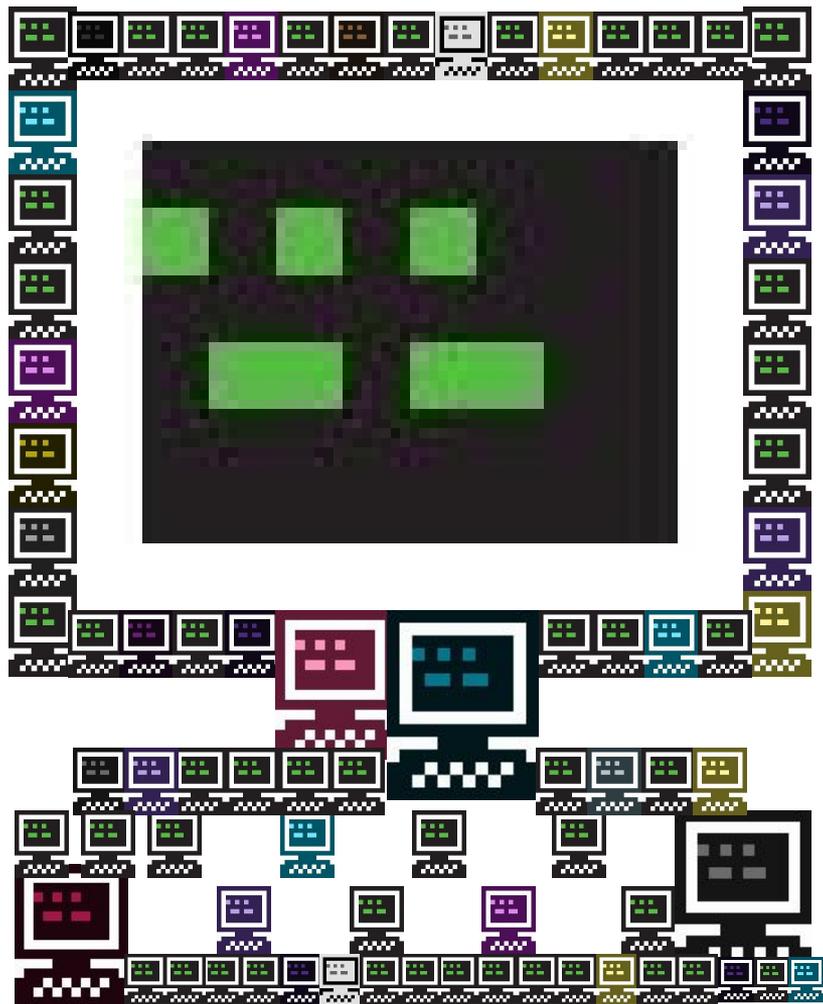


**object-  
oriented**



"Slide of the Week: Increased WBC, April 14, 2011" via Center for Genomic Pathology  
(<http://ctrngenpath.net/wp-content/uploads/2011/04/cml02.jpg>)





# Alan Kay

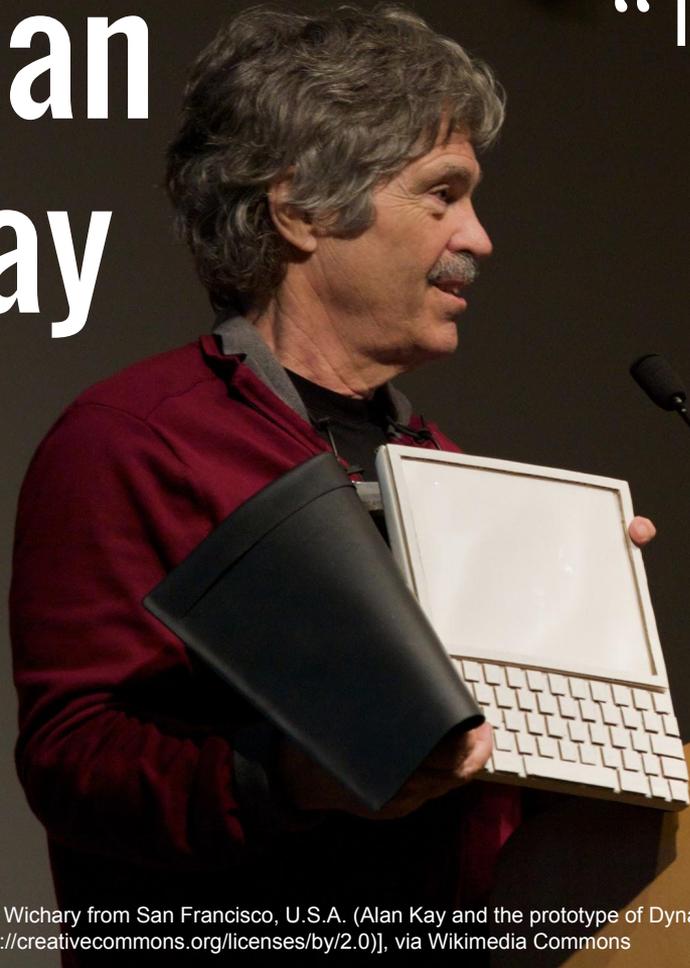
“I'm sorry that I long ago coined the term "objects" for this topic because it gets many people to focus on the lesser idea.”

# Alan Kay

“The big idea is  
"messaging"”

- To Smalltalk/Squeak  
mailing list, 1998

<http://wiki.c2.com/?AlanKayOnMessaging>



oops #2:

big ideas

**what I  
thought**

**objects**

**classes**

---

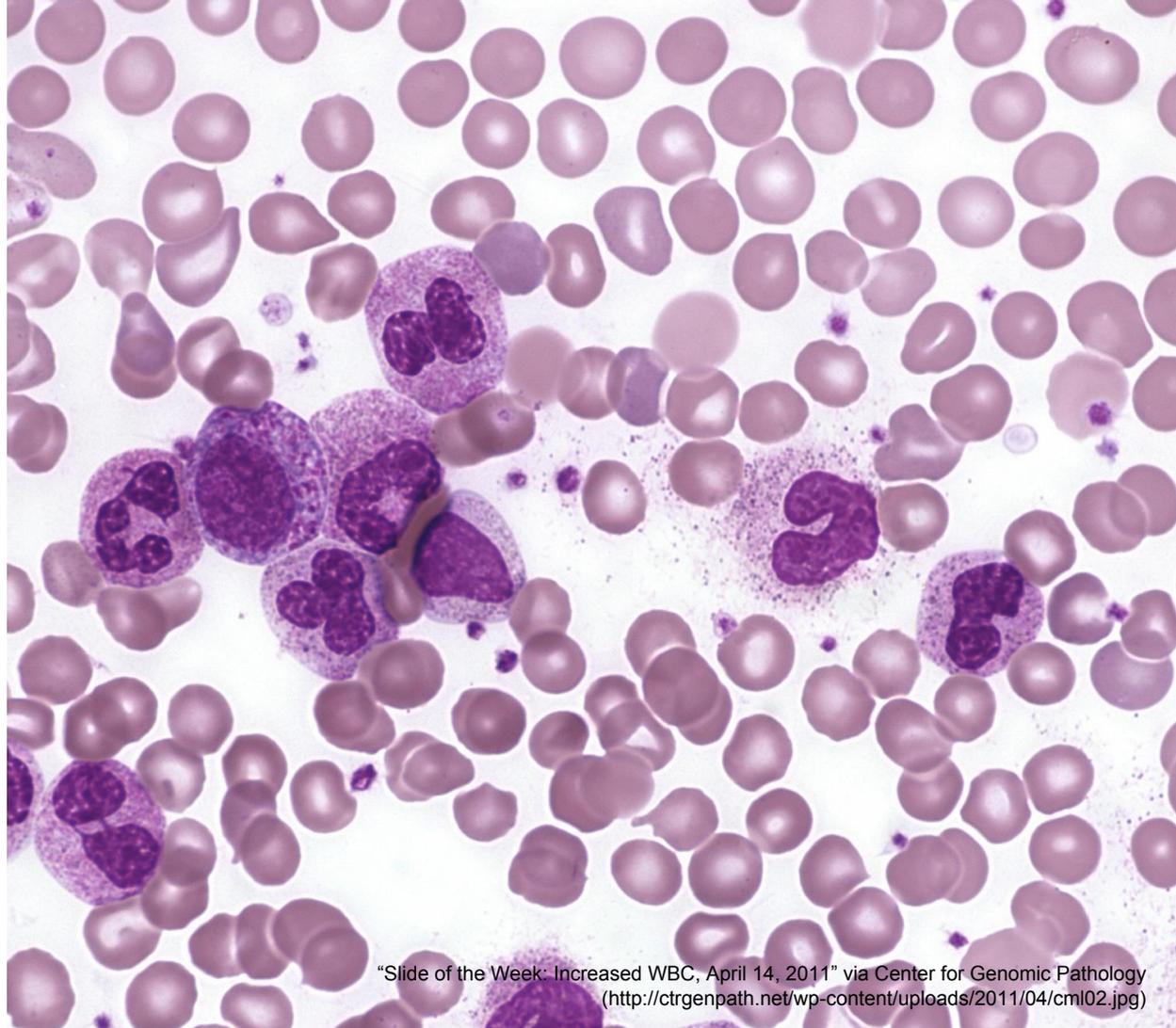
**what it is**

**messaging**

**late-binding**

---

**objects**



"Slide of the Week: Increased WBC, April 14, 2011" via Center for Genomic Pathology  
(<http://ctrngenpath.net/wp-content/uploads/2011/04/cml02.jpg>)

**messages**



"Dopamine D3 receptor" via Orientation of Proteins in Membranes Database  
(<http://opm.phar.umich.edu/images/png/3pbl.png>)

**systems**

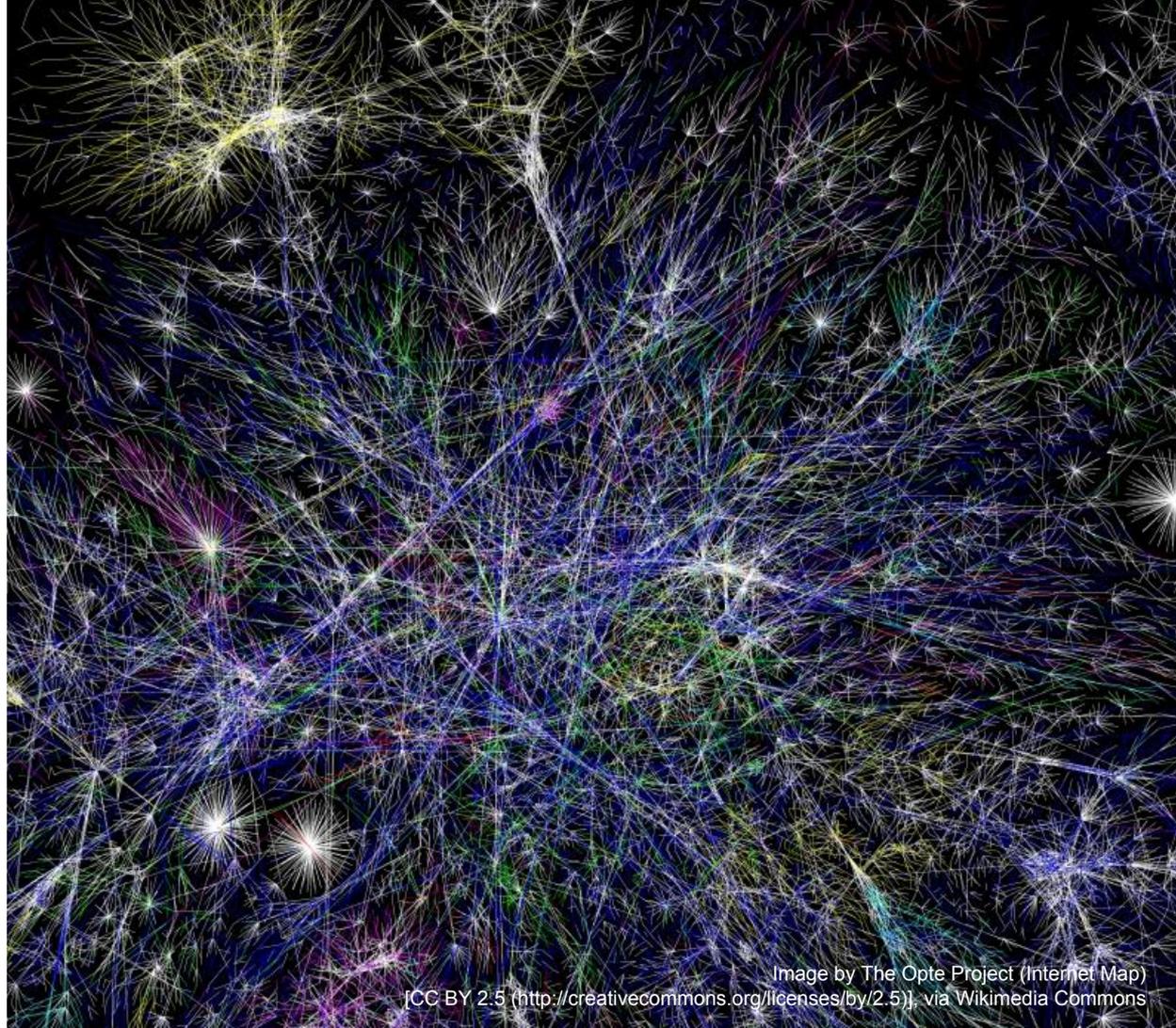
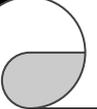


Image by The Opte Project (Internet Map)  
[CC BY 2.5 (<http://creativecommons.org/licenses/by/2.5>)], via Wikimedia Commons

```
thing.do(some, stuff)
```



to: thing

subject: pretty please? :D

message: "do", some, stuff



recipient



message



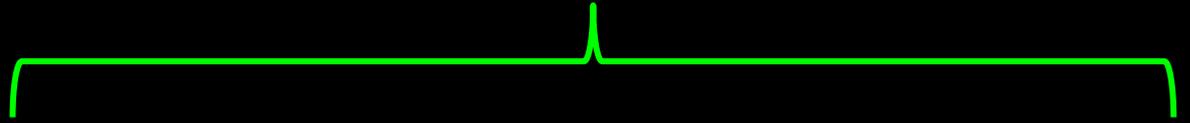
thing.do(some, stuff)

recipient



thing.do(some, stuff)

message



method name



arguments



# Ruby

```
thing.do(some, stuff)
```

```
thing.send(:do, some, stuff)
```

# what does **do** do?

thing.**do**(some, stuff)

thing.**do**(**other**, stuff)

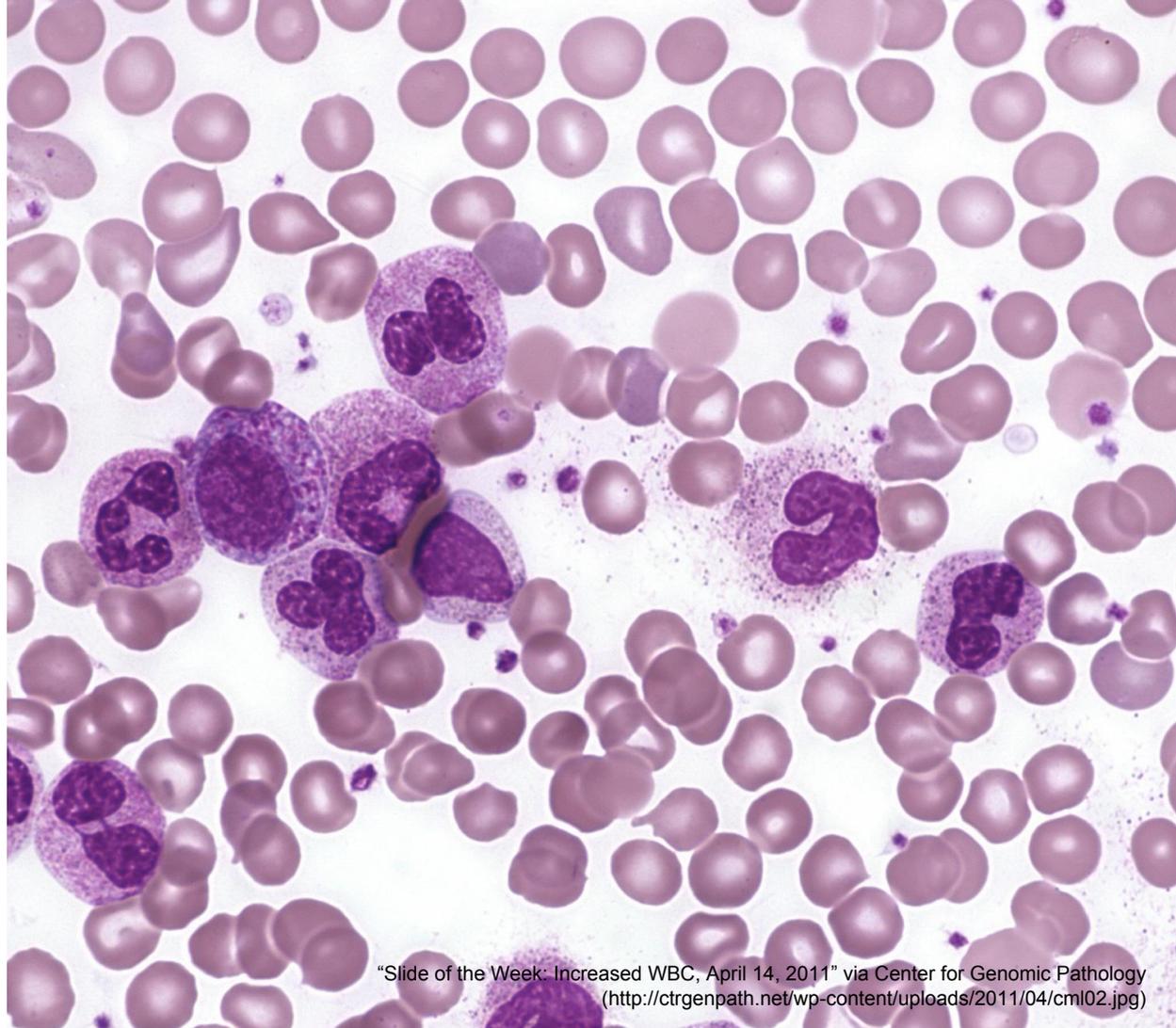
**child**.**do**(some, stuff)

thing.**do**(some, stuff)

**early  
binding**



**late  
binding**



**oops #3:**

**beings vs. behaviors**

**what I  
thought**

**objects  
are nouns**

---

**what it is**

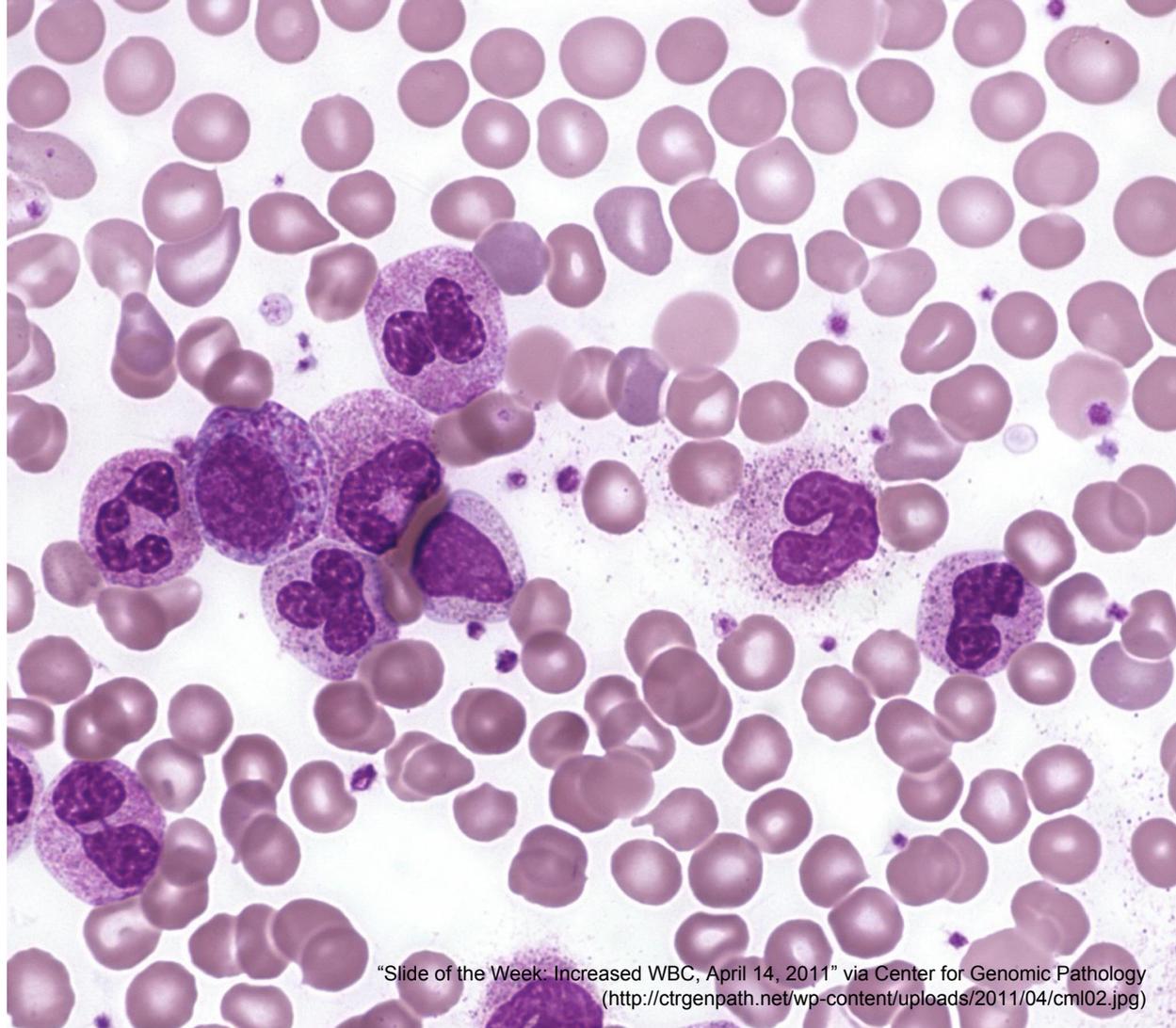
**objects  
are verbs**

---

**data**



# behavior



"Slide of the Week: Increased WBC, April 14, 2011" via Center for Genomic Pathology  
(<http://ctrngenpath.net/wp-content/uploads/2011/04/cml02.jpg>)

`thing.do(stuff)`  
=> behavior

```
class Friend:
    def __init__(self, friends):
        self.friends = friends
    def is_friend_of(self, name):
        return name in self.friends
```

```
buddy = Friend(['alan', 'alonzo'])
buddy.is_friend_of('guy') => False
```

```
buddy.is_friend_of('guy')
```

```
buddy.send(:is_friend_of, 'guy')
```

```
buddy.is_friend_of('guy')
```

```
buddy.send(:is_friend_of, 'guy')
```

```
buddy('is_friend_of', 'guy')
```

```
buddy('is_friend_of', 'guy') => False
```

```
def buddy(method_name, *args):  
    if method_name == 'is_friend_of':  
        return is_my_friend(*args)
```

buddy('is\_friend\_of', 'guy') => False

```
def is_my_friend(name):  
    return name in friend_names  
def buddy(method_name, *args):  
    if method_name == 'is_friend_of':  
        return is_my_friend(*args)
```

buddy('is\_friend\_of', 'guy') => False

```
def Friend(friend_names):
    my_friends = friend_names
    def is_my_friend(name):
        return name in friend_names
    def responder(method_name, *args):
        if method_name == 'is_friend_of':
            return is_my_friend(*args)
    return responder
```

```
buddy = Friend(['alan', 'alonzo'])
buddy('is_friend_of', 'guy') => False
```

**oops #4:**

**beginnings**

# what I thought

60's-80's

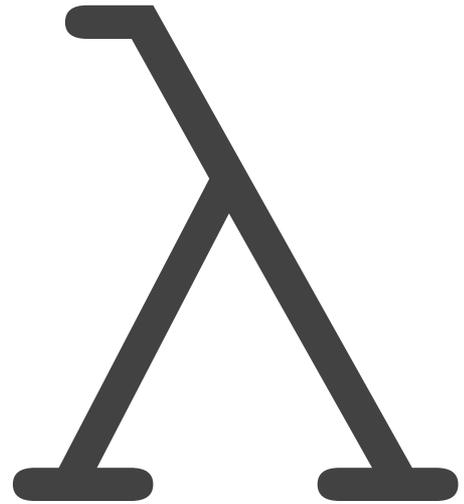
Simula

Smalltalk

C++

---

what it is



# Lambda calculus

TRUE :=  $\lambda x.\lambda y.x$

FALSE :=  $\lambda x.\lambda y.y$

# Smalltalk

```
class True
  ifTrue: a ifFalse: b
    ^ a value
```

```
class False
  ifTrue: a ifFalse: b
    ^ b value
```

oops #5:

boundaries

what I  
thought

**imperative**

-----

**oo**

=====

**functional**

---

what it is

**imperative**

=====

**oo**

-----

**functional**

---

oops #6:

**belief structures**

**what I  
thought**

**dogmatic &  
restrictive**

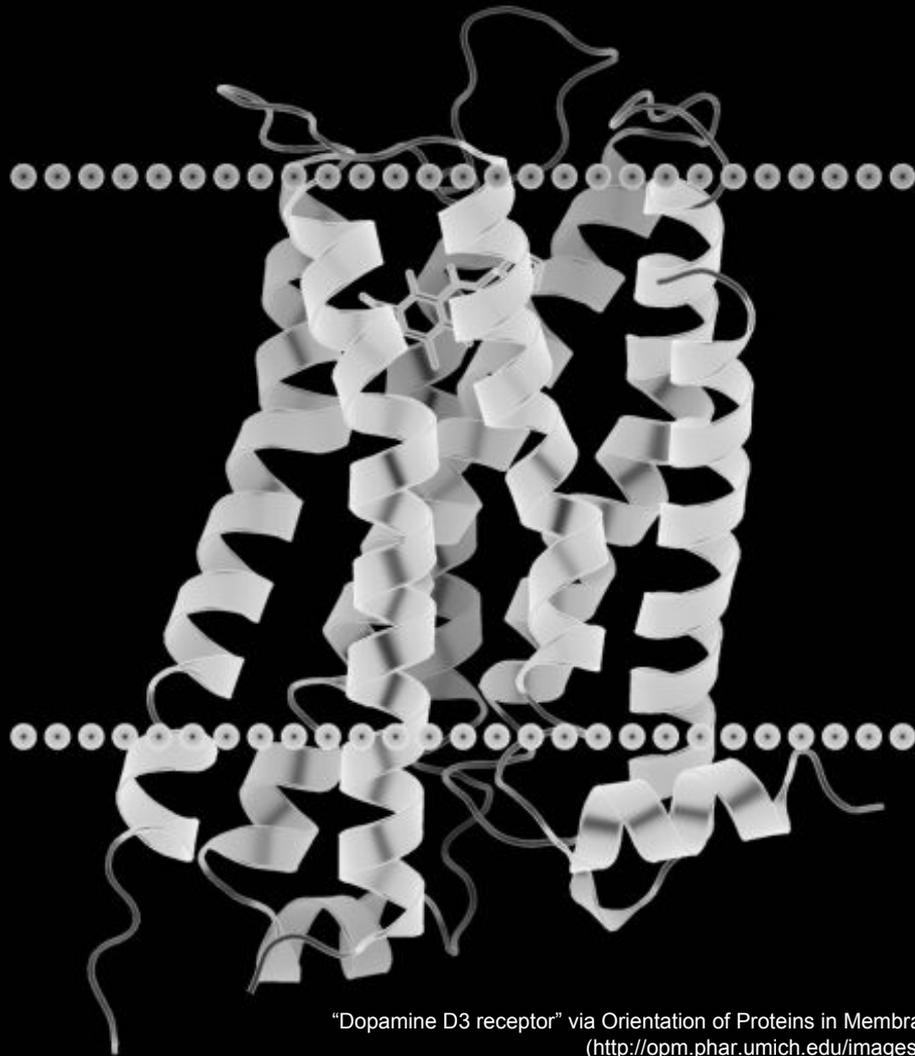
---

**what it is**

**built for  
change**

**---  
\*or should be**

**simplicity**  
**complexity**



"Dopamine D3 receptor" via Orientation of Proteins in Membranes Database  
(<http://opm.phar.umich.edu/images/png/3pbl.png>)

# scalability

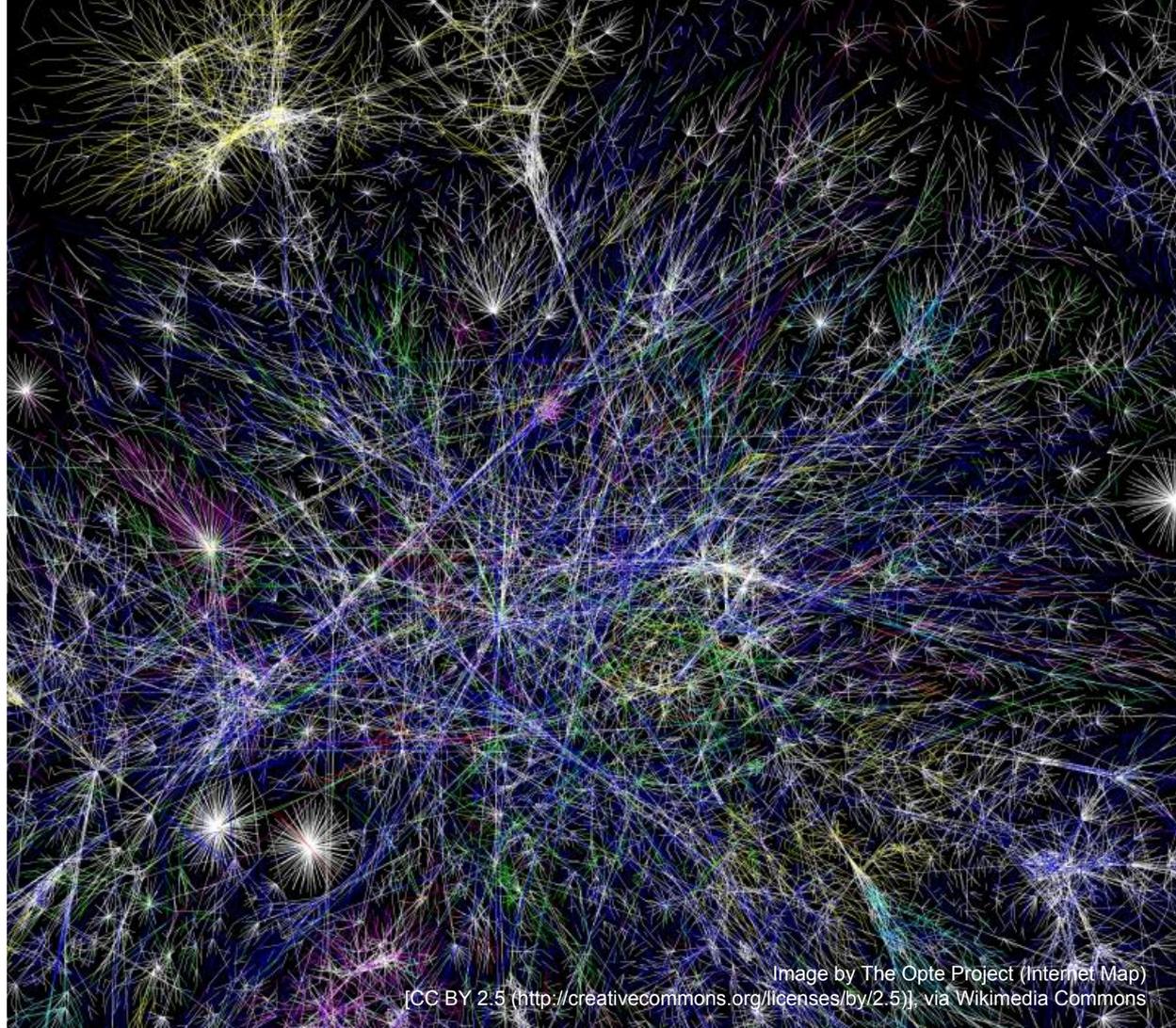
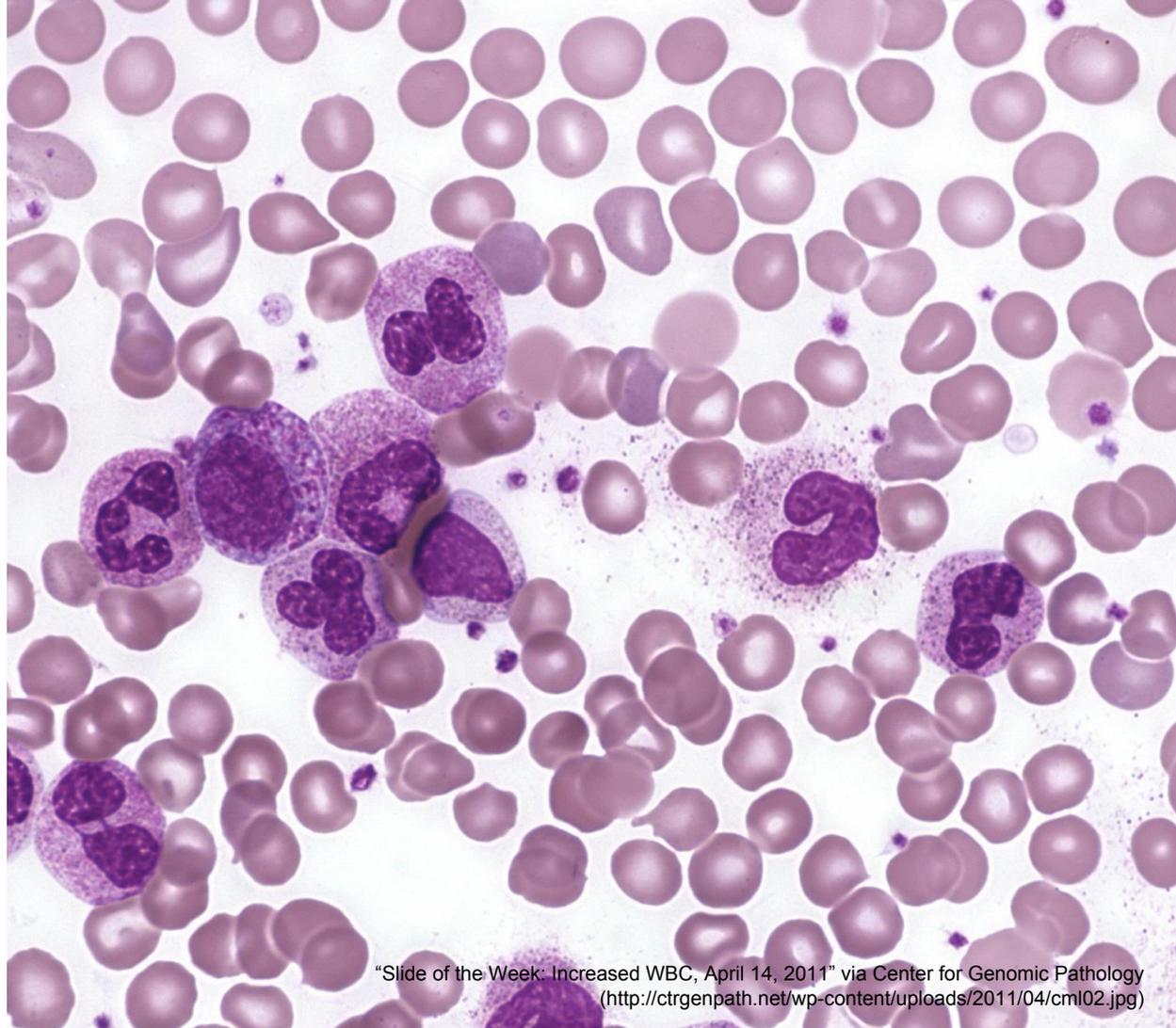


Image by The Opte Project (Internet Map)  
[CC BY 2.5 (<http://creativecommons.org/licenses/by/2.5>)], via Wikimedia Commons

**resilience**



"Slide of the Week: Increased WBC, April 14, 2011" via Center for Genomic Pathology  
(<http://ctrngenpath.net/wp-content/uploads/2011/04/cml02.jpg>)

## Alan Kay:

- The early history of Smalltalk [worrydream.com/EarlyHistoryOfSmalltalk](http://worrydream.com/EarlyHistoryOfSmalltalk)
- Comment on Moti Ben-Ari's "Objects never? Well, hardly ever!"  
[computinged.wordpress.com/2010/09/11/moti-asks-objects-never-well-hardly-ever/#div-comment-3766](http://computinged.wordpress.com/2010/09/11/moti-asks-objects-never-well-hardly-ever/#div-comment-3766)
- E-mail to Squeak list on "messaging" [wiki.c2.com/?AlanKayOnMessaging](http://wiki.c2.com/?AlanKayOnMessaging)

## William Cook:

- A Proposal for Simplified, Modern Definitions of "Object" and "Object Oriented"  
[wcook.blogspot.com/2012/07/proposal-for-simplified-modern.html](http://wcook.blogspot.com/2012/07/proposal-for-simplified-modern.html)
- On Understanding Data Abstraction, Revisited  
[www.cs.utexas.edu/%7Ewcook/Drafts/2009/essay.pdf](http://www.cs.utexas.edu/%7Ewcook/Drafts/2009/essay.pdf)

**Structure & Interpretation of Computer Programs** [mitpress.mit.edu/sicp](http://mitpress.mit.edu/sicp)

**David Albert, Darius Bacon, Lindsey Kuper,  
Julia Evans, Matthew Parker, Ezekiel Smithburg,  
Juan Hernández, Andy Lee, Michael Arntzenius  
& the Recusers  
Team DomCode**

**thank you!**

**@AnjanaVakil**